Linear Maximum Margin Classifier for Learning from Uncertain Data

Christos Tzelepis, *Student Member, IEEE,* Vasileios Mezaris, *Senior Member, IEEE,* and Ioannis Patras, *Senior Member, IEEE*

***Abstract*—In this paper, we propose a maximum margin classifier that deals with uncertainty in data input. Specifically, we reformulate the SVM framework such that each input training entity is not solely a feature vector representation, but a multi-dimensional Gaussian distribution with given probability density, i.e., with a given mean and covariance matrix. The latter expresses the uncertainty. We arrive at a convex optimization problem, which is solved in the primal form using a gradient descent approach. The resulting classifier, which we name SVM with Gaussian Sample Uncertainty (SVM-GSU), is tested on synthetic data, as well as on the problem of event detection in video using the large-scale TRECVID MED 2014 dataset, and the problem of image classification using the MNIST dataset of handwritten digits. Experimental results verify the effectiveness of the proposed classifier.**

***Index Terms*—Classification, convex optimization, Gaussian anisotropic uncertainty, large margin methods, learning with uncertainty, statistical learning theory**

I. Introduction

SUPPORT Vector Machine (SVM) has been shown to be a powerful paradigm for pattern classification. The origins of SVM can be traced back to [1], [2]. In [3], Vapnik established the standard regularized SVM algorithm where a linear discriminative function is computed in order to achieve maximum sample margin. To this end, a penalty term approximating the total training error is considered along with a regularization term, typically chosen as a norm of the classifier, in order to avoid the so-called over-fitting phenomenon. From a statistical learning theory point of view, this is interpreted as follows: the regularization term restricts the complexity of the classifier and thus the deviation of the testing error. Hence, the training error is controlled (see e.g. [4], [5], [6]). The training data are assumed to be drawn from some unknown probability distribution; specifically, they are assumed to be independently drawn and identically distributed ("iid").

The majority of the classification methods do not address the uncertainty in the training data explicitly. That is, each training sample is described by its position in some vector space (feature representation). However, such an approach often does not express the true underlying process of extracting the feature representation. Errors are often introduced during sensing or feature extraction and therefore the training data are noisy. In this work, we model the uncertainty of each training example using a multivariate Gaussian distribution, such that the covariance matrix of each distribution is treated as a measure of this uncertainty. That is, we model each input example as a random vector following a multivariate Gaussian distribution with given mean vector and covariance matrix. In Fig.1 we can see such 2D training examples, given as bivariate Gaussian distributions with certain mean vectors and covariance matrices. For the sake of visualization, we illustrate the uncertainty of each input training vector with the shaded regions, which are bounded by the iso-density loci of points (ellipses) described by the 0.03% of the maximum density of each distribution. A novel SVM formulation is developed, by modifying appropriately the mechanism for measuring the classification (empirical) error and for taking it into account during training. Hereafter, the proposed algorithm will be called SVM with Gaussian Sample Uncertainty (SVM-GSU). The toy example in Fig.1 illustrates the motivation behind the proposed SVM-GSU. That is, the decision boundary of the SVM-GSU, shown with a solid line, may be drastically different than that of the standard SVM, shown with a dashed line, when taking into account the uncertainty associated with each input data.

The remainder of this paper is organized as follows. In

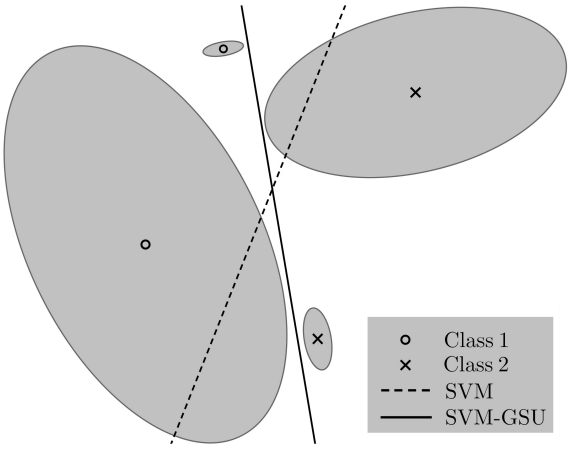


Fig. 1: Linear SVM with Gaussian Sample Uncertainty (SVM-GSU). Solid line illustrates the decision boundary of the proposed algorithm, and dashed line shows the decision boundary of the standard linear SVM.

C. Tzelepis is with the Information Technologies Institute/Centre for Research and Technology Hellas (CERTH), Thermi 57001, Greece, and also with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, U.K. (email: tzelepis@iti.gr).

V. Mezaris is with the Information Technologies Institute/Centre for Research and Technology Hellas (CERTH), Thermi 57001, Greece (email: bmezaris@iti.gr).

I. Patras is with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, U.K. (e-mail: i.patras@qmul.ac.uk).

Section II, we review related work. In Section III, we present the proposed SVM-GSU. In Section IV, we provide the experimental results of the application of SVM-GSU to synthetic data, the TRECVID MED 2014 dataset, as well as the MNIST dataset, along with comparisons with the standard SVM, and other state of the art methods. We discuss conclusions in Section V.

II. Related Work

Assuming uncertainty in input under the SVM paradigm is not new. Different types of Robust SVMs have been proposed in several recent works. Bi and Zhang [7] considered a statistical formulation where the input noise is modeled as a hidden mixture component, but in this way the "iid" assumption for the training data is violated. In that work, the uncertainty is modeled isotropically. Second order cone programming (SOCP) [8] methods have also been employed in numerous works to handle missing and uncertain data. In addition, Robust Optimization [9], [10] techniques have been proposed for optimization problems where data is not specified exactly, but it is known to belong to a given uncertainty set $\mathcal{U}$, yet the constraints of the optimization problem must hold for all possible values of the data from $\mathcal{U}$.

Lanckriet et al. [11] considered a binary classification problem where the mean and covariance matrix of each class are assumed to be known. Then, a minimax problem is formulated such that the worst-case (maximum) probability of misclassification of future data points is minimized. That is, under all possible choices of class-conditional densities with a given mean and covariance matrix, the worst-case probability of misclassification of new data is minimized. For doing so, the authors exploited generalized Chebyshev inequalities [12] and particularly a theorem according to which the probability of misclassifying a point is bounded.

Shivaswamy et al. [13], who extended Bhattacharyya et al. [14], also adopted a second order cone programming formulation and used generalized Chebyshev inequalities to design robust classifiers dealing with uncertain observations. Then uncertainty arises in ellipsoidal form, as follows directly from the multivariate Chebyshev inequality. This formulation achieves robustness by requiring that the ellipsoid of every uncertain data point should lie in the correct half-space. The expected error of misclassifying a sample is obtained by computing the volume of the ellipsoid that lies on the wrong side of the hyperplane. However, this quantity is not computed analytically; instead, a large number of uniformly distributed points are generated in the ellipsoid, and the fraction of the number of points on the wrong side of the hyperplane to the total number of generated points is computed.

Xu et al. [15], [16] considered the robust classification problem for a class of non-box-typed uncertainty sets, in contrast to [14], [13], [11], who robustified regularized classification using box-type uncertainty. That is, they considered a setup where the joint uncertainty is the Cartesian product of uncertainty in each input, leading to penalty terms on each constraint of the resulting formulation. Furthermore, Xu et al. gave evidence on the equivalence between the standard regularized SVM and this robust optimization formulation, establishing robustness as the *reason* why regularized SVMs generalize well.

In [17], motivated by GEPSVM [18], Qi et al. robustified a twin support vector machine (TWSVM) [19]. Robust TWSVM deals with data affected by measurement noise using a second order cone programming formulation. In their work, input data is contaminated with isotropic noise (i.e., spherical disturbances centred at the training samples), and thus cannot model real-world uncertainty, which is typically described by more complex noise patterns. Our proposed classifier, which is presented below, does not violate the "iid" assumption for the training input data, while it can model the uncertainty of each input training example using an arbitrary covariance matrix, consequently permitting the uncertainty to be anisotropic. Moreover, the expected error is computed analytically and is minimized by an iterative gradient descent algorithm whose complexity is linear with respect to the number of training data. Finally, we apply a linear subspace learning approach in order to solve the problem in lower-dimensional spaces, and thus accelerate the training stage. Learning in subspaces is widely used in various statistical learning problems [20], [21], [22], [23].

III. Proposed Approach

As discussed above, in this section we develop a new algorithm, in which the training set that feeds the proposed classifier includes training examples described not solely by a set of feature representations, i.e. a set of vectors $\mathbf{x}_i$ in some n-dimensional space, but rather by a set of multivariate Gaussian distributions; that is, every training data is characterized by a mean vector $\mathbf{x}_i \in \mathcal{D}$ and a covariance matrix $\Sigma_i \in \mathbb{S}^n_{++}$*. A linear formulation is proposed below, while an approximation formulation dealing with learning in linear subspaces is discussed next.

A. SVM with Gaussian Sample Uncertainty (SVM-GSU)

Let us briefly begin with the baseline SVM algorithm, which will endow us with arguments necessary for generalizing and proceeding to the proposed approach. We consider the supervised learning framework where a set of l annotated observations is available. That is, each observation consists of a vector, $\mathbf{x}_i$, in some n-dimensional vector space, let $D \subseteq \mathbb{R}^{n\dagger}$, and an associated label, $y_i \in \{\pm 1\}$. Let us denote the training set by $\mathcal{X} = \big\{(\mathbf{x}_i, y_i) \colon \mathbf{x}_i \in \mathbb{R}^n,\ y_i \in \{\pm 1\}, i = 1, \ldots, l\big\}$. Then, the baseline linear SVM [3] learns a hyperplane $\mathcal{H} \colon \mathbf{w} \cdot \mathbf{x} + b = 0$ that minimizes with respect to $\mathbf{w}$, b the following objective function

$$\frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{l} \max(0, 1 - y_i(\mathbf{w} \cdot \mathbf{x}_i + b)), \qquad (1)$$

where $h(y, t) = \max(0, 1 - yt)$ is known as the "hinge loss" function [24].

*$\mathcal{D}$ is typically a subset of the n-dimensional Euclidean space of column vectors, while $\mathbb{S}^n_{++}$ denotes the convex cone of all symmetric positive definite $n \times n$ matrices with entries in $\mathcal{D} \subseteq \mathbb{R}^n$.

†For the rest of this paper, we will assume that $\mathcal{D} \equiv \mathbb{R}^n$.

In this work, we assume that instead of the i-th training example, we are given a multivariate Gaussian distribution with mean vector $\mathbf{x}_i$, and covariance matrix Σ_i. One could think of this as that the covariance matrix, Σ_i, describes the uncertainty about the position of the training sample around $\mathbf{x}_i$. Formally, we define random variables, $\mathbf{X}_i$, each of which follows an n-dimensional Gaussian distribution with mean vector $\mathbf{x}_i \in \mathbb{R}^n$, and covariance matrix a symmetric positive definite $n \times n$ matrix, $\Sigma_i \in \mathbb{S}^n_{++}$. The probability density function (pdf) of the i-th Gaussian distribution is given by $f_{\mathbf{X}_i}\colon \mathbb{R}^n \to \mathbb{R}$, with

$$f_{\mathbf{X}_i}(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma_i|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{x}_i)^\top \Sigma_i^{-1}(\mathbf{x}-\mathbf{x}_i)\right). \tag{2}$$

Adopting the above assumption for the input training vectors, we can express the training set as a set of l annotated Gaussian distributions, i.e., $\mathcal{X}' = \big\{(\mathbf{x}_i, \Sigma_i, y_i)\colon \mathbf{x}_i \in \mathbb{R}^n, \Sigma_i \in \mathbb{S}^n_{++}, y_i \in \{\pm 1\},\ i = 1, \ldots, l\big\}$. The optimization problem, in its unconstrained form, is then formulated as follows

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l} \int_{\mathbb{R}^n} \max\big(0, 1 - y_i(\mathbf{w}\cdot\mathbf{x}+b)\big) f_{\mathbf{X}_i}(\mathbf{x})\,\mathrm{d}\mathbf{x}, \tag{3}$$

or,

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l} \int_{\Omega_i} \big[1 - y_i(\mathbf{w}\cdot\mathbf{x}+b)\big] f_{\mathbf{X}_i}(\mathbf{x})\,\mathrm{d}\mathbf{x}, \tag{4}$$

where Ω_i denotes the half-space of $\mathbb{R}^n$ that is defined by the hyperplane $\mathcal{H}'\colon y_i(\mathbf{w}\cdot\mathbf{x}+b) = 1$ as $\Omega_i = \{\mathbf{x} \in \mathbb{R}^n\colon y_i(\mathbf{w}\cdot\mathbf{x}+b) \leq 1\}$, and is the half-space to which a misclassified sample lies.

Note that the loss function $\mathcal{L}\colon \big(\mathbb{R}^n \times \mathbb{R}\big) \times \big(\mathbb{R}^n \times \mathbb{S}^n_{++} \times \{\pm 1\}\big) \to \mathbb{R}$ that can be defined for the samples drawn from the i-th Gaussian, that is,

$$\mathcal{L}(\mathbf{w}, b, \mathbf{x}_i, \Sigma_i, y_i) = \int_{\Omega_i} \big[1 - y_i(\mathbf{w}\cdot\mathbf{x}+b)\big] f_{\mathbf{X}_i}(\mathbf{x})\,\mathrm{d}\mathbf{x}, \tag{5}$$

is the expected value of the hinge loss. Using the Theorem 1 proved in Appendix A, for the half-spaces $\Omega_i^+ = \big\{\mathbf{x} \in \mathbb{R}^n\colon \mathbf{w}\cdot\mathbf{x}+b-1 \geq 0\big\}$, and $\Omega_i^- = \big\{\mathbf{x} \in \mathbb{R}^n\colon \mathbf{w}\cdot\mathbf{x}+b-1 \leq 0\big\}$, the above integral is evaluated in terms of $\mathbf{w}$ and b as follows

$$\begin{aligned}\mathcal{L}(\mathbf{w}, b, \mathbf{x}_i, \Sigma_i, y_i) = \\ \frac{1 - y_i(\mathbf{w}\cdot\mathbf{x}_i+b)}{2}\left[y_i \operatorname{erf}\left(\frac{y_i - (\mathbf{w}\cdot\mathbf{x}_i+b)}{\sqrt{2\mathbf{w}^\top\Sigma_i\mathbf{w}}}\right) + 1\right] \\ + \frac{\sqrt{\mathbf{w}^\top\Sigma_i\mathbf{w}}}{\sqrt{2\pi}} \exp\left(-\frac{[y_i - (\mathbf{w}\cdot\mathbf{x}_i+b)]^2}{2\mathbf{w}^\top\Sigma_i\mathbf{w}}\right),\end{aligned} \tag{6}$$

where $\operatorname{erf}\colon \mathbb{R} \to (-1, 1)$ is the error function, defined as

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\,\mathrm{d}t.$$

As stated above, the covariance matrix of each training random vector describes its uncertainty, and as the covariance matrix approaches to the zero matrix, the certainty increases. At the extreme, as $\Sigma_i \to \mathbf{0}$, after applying function analysis, (6) yields $1 - y_i(\mathbf{w}\cdot\mathbf{x}_i + b)$, which is the hinge loss function used in the standard SVM formulation [3], [25], [24]. That implies that the proposed formulation is a generalization of the standard SVM; the two classifiers are equivalent when the covariance matrices tend to the zero matrix[‡].

Let $\mathcal{J}\colon \big(\mathbb{R}^n \times \mathbb{R}\big) \times \big(\mathbb{R}^n \times \mathbb{S}^n_{++} \times \{\pm 1\}\big) \to \mathbb{R}$ be the objective function of the SVM-GSU formulation, i.e.,

$$\mathcal{J}(\mathbf{w}, b, \mathbf{x}_i, \Sigma_i, y_i) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l} \mathcal{L}(\mathbf{w}, b, \mathbf{x}_i, \Sigma_i, y_i), \tag{7}$$

which is convex as proved in Appendix B.

To solve the convex optimization problem (4), the Limited-memory BFGS (L-BFGS) algorithm has been employed[§]. L-BFGS belongs to the family of quasi-Newton methods and approximates the BFGS algorithm [26] using a limited amount of memory. L-BFGS requires the first-order derivatives with respect to the optimization variables $\mathbf{w}$, b. Then, the objective function is minimized jointly for $\mathbf{w}$, b and a (global) optimal solution is achieved. By differentiating $\mathcal{J}$ with respect to $\mathbf{w}$ and b, we obtain, respectively,

$$\begin{aligned}\frac{\partial}{\partial\mathbf{w}}\mathcal{J}(\mathbf{w}, b) = \mathbf{w} + C\sum_{i=1}^{l}\Bigg[\frac{\exp\left(-\frac{[y_i-(\mathbf{w}\cdot\mathbf{x}_i+b)]^2}{2\mathbf{w}^\top\Sigma_i\mathbf{w}}\right)}{\sqrt{2\pi\mathbf{w}^\top\Sigma_i\mathbf{w}}}\Sigma_i\mathbf{w} \\ -\frac{1}{2}\left(\operatorname{erf}\left(\frac{y_i - (\mathbf{w}\cdot\mathbf{x}_i+b)}{\sqrt{2\mathbf{w}^\top\Sigma_i\mathbf{w}}}\right) + y_i\right)\mathbf{x}_i\Bigg],\end{aligned} \tag{8}$$

$$\frac{\partial}{\partial b}\mathcal{J}(\mathbf{w}, b) = -\frac{C}{2}\sum_{i=1}^{l}\left[\operatorname{erf}\left(\frac{y_i - (\mathbf{w}\cdot\mathbf{x}_i+b)}{\sqrt{2\mathbf{w}^\top\Sigma_i\mathbf{w}}}\right) + y_i\right]. \tag{9}$$

By applying L-BFGS on the problem of (4), we obtain the optimal values of the parameters $\mathbf{w}$, b defining the SVM-GSU's learned separating hyperplane.

Then, given this hyperplane $\mathcal{H}\colon \mathbf{w}\cdot\mathbf{x} + b = 0$, an unseen testing datum, $\mathbf{x}_t$, is classified to one of the two classes according to the sign of the (signed) distance between $\mathbf{x}_t$ and the separating hyperplane. That is, the predicted label of $\mathbf{x}_t$ is computed as $y_t = \operatorname{sgn}(d_t)$, where $d_t = (\mathbf{w}\cdot\mathbf{x}_t+b)/\|\mathbf{w}\|$, while a probabilistic degree of confidence (DoC) that the testing sample belongs to the class to which it has been classified can be calculated using the well-known sigmoid function, $S(d_t) = 1/(1+e^{-d_t})$. This is the same approach that is used in the baseline linear SVM formulation [27] for evaluating a sample's class membership at the testing phase.

B. Solving the SVM-GSU in linear subspaces

Since learning in the original n-dimensional input space may introduce computationally expensive terms, in this section we propose a methodology for approximating the loss function of SVM-GSU, by projecting each input random vector into a linear subspace. The dimensionality of each subspace is defined by preserving a given fraction of the total variance for each covariance matrix. Then, the total loss, as well as

[‡]A zero covariance matrix exists due to the well known property that the set of symmetric positive definite matrices is a convex cone with vertex at zero.

[§]A framework for training and testing the linear SVM-GSU has been developed in C and is publicly available at <withheld during reviewing>.

its first derivatives, are computed separately in each subspace. A comprehensive analysis of the above method is discussed below.

By performing eigenanalysis in the covariance matrix of $\mathbf{X}_i$, the latter is decomposed as follows

$$\Sigma_i = U_i \Lambda_i U_i^\top, \tag{10}$$

where Λ_i is an $n \times n$ diagonal matrix consisting of the eigenvalues of Σ_i, i.e. $\Lambda_i = \mathrm{diag}(\lambda_1^i, \ldots, \lambda_n^i)$, such that $\lambda_1^i \geq \lambda_2^i \geq, \ldots, \geq \lambda_n^i > 0$, while U_i is an $n \times n$ orthonormal matrix, whose j-th column, $\mathbf{u}_i^j$, is the eigenvector corresponding to the j-th eigenvalue, λ_j^i. Let us keep the first $d_i \leq n$ eigenvectors, such that a certain percentage ϵ (e.g. $\epsilon = 90\%$) of the total variance is preserved, i.e.

$$\frac{\sum_{t=1}^{d_i} \lambda_t^i}{\sum_{t=1}^{n} \lambda_t^i} > \epsilon.$$

Then, we construct the $n \times d_i$ matrix U_i' by keeping the first d_i columns of U_i, i.e.,

$$U_i' = [\mathbf{u}_1^i \;\; \mathbf{u}_2^i \;\; \ldots \;\; \mathbf{u}_{d_i}^i] \in \mathbb{R}^{n \times d_i}. \tag{11}$$

Now, by using the matrix $P_i = U_i'^\top \in \mathbb{R}^{d_i \times n}$, we define a new random vector $\mathbf{Z}_i$, such that

$$\mathbf{Z}_i = P_i \mathbf{X}_i. \tag{12}$$

Then, $\mathbf{Z}_i \in \mathbb{R}^{d_i}$ follows a multivariate Gaussian distribution (since $\mathbf{X}_i \sim \mathcal{N}(\mathbf{x}_i, \Sigma_i)$), i.e. $\mathbf{Z}_i \sim \mathcal{N}(\mathbf{z}_i, \Sigma_i^z)$, with mean vector

$$\mathbf{z}_i = \mathbb{E}\big[P_i \mathbf{X}_i\big] = P_i \mathbb{E}\big[\mathbf{X}_i\big] = P_i \mathbf{x}_i \in \mathbb{R}^{d_i}, \tag{13}$$

and covariance matrix

$$\Sigma_i^z = \Lambda_i^z = \mathrm{diag}\big(\lambda_1, \ldots, \lambda_{d_i}\big). \tag{14}$$

The probability density function of $\mathbf{Z}_i$ is given by $f_{\mathbf{Z}_i} \colon \mathbb{R}^{d_i} \to \mathbb{R}$, with

$$f_{\mathbf{Z}_i}(\mathbf{z}) = \frac{1}{(2\pi)^{\frac{d_i}{2}} |\Sigma_i^z|^{\frac{1}{2}}} \exp\Big(-\frac{1}{2} (\mathbf{z} - \mathbf{z}_i)^\top \Sigma_i^{z\,-1} (\mathbf{z} - \mathbf{z}_i) \Big). \tag{15}$$

$P_i^\top$ is a projection matrix from $\mathbb{R}^n$ to the d_i-dimensional space $\mathbb{R}^{d_i}$. Let us now see how the integral in (4) is approximated in the new space. To this end, the following holds true

$$\mathbf{w} \cdot \mathbf{x} \approx \mathbf{w}^\top (P_i^\top \mathbf{z}) = (P_i^\top \mathbf{z})^\top \mathbf{w} = \mathbf{z}^\top P_i \mathbf{w} = (P_i \mathbf{w}) \cdot \mathbf{z},$$

or, by letting $\mathbf{w}_z = P_i \mathbf{w}$,

$$\mathbf{w} \cdot \mathbf{x} \approx \mathbf{w_z} \cdot \mathbf{z}.$$

Consequently, the integral in the RHS of (4) can be approximated by the quantity

$$\int_{\Omega_i^z} \Big[1 - y_i\big(\mathbf{w}_z \cdot \mathbf{z} + b\big)\Big] f_{\mathbf{Z}_i}(\mathbf{z})\, \mathrm{d}\mathbf{z},$$

where Ω_i^z denotes the projected half-space on $\mathbb{R}^{d_i}$, that is,

$$\Omega_i^z = \Big\{ \mathbf{z} \in \mathbb{R}^{d_i} \colon y_i\big(\mathbf{w}_z \cdot \mathbf{z} + b\big) \leq 1 \Big\}.$$

Using Theorem 1, which is proved in Appendix A, the above integral is equal to

$$\frac{1 - y_i(\mathbf{w}_z \cdot \mathbf{z}_i + b)}{2} \left[y_i \operatorname{erf}\left(\frac{y_i - (\mathbf{w}_z \cdot \mathbf{z}_i + b)}{\sqrt{2 \mathbf{w}_z^\top \Sigma_i^z \mathbf{w}_z}} \right) + 1 \right] + \frac{\sqrt{\mathbf{w}_z^\top \Sigma_i^z \mathbf{w}_z}}{\sqrt{2\pi}} \exp\left(-\frac{[y_i - (\mathbf{w}_z \cdot \mathbf{z}_i + b)]^2}{2 \mathbf{w}_z^\top \Sigma_i^z \mathbf{w}_z} \right). \tag{16}$$

Therefore, for each training example (i.e., for each random vector that follows a given Gaussian distribution), the loss function $\mathcal{L}_i^z \colon \big(\mathbb{R}^{d_i} \times \mathbb{R}\big) \times \big(\mathbb{R}^{d_i} \times \mathbb{S}_{++}^{d_i} \times \{\pm 1\}\big)$, is given by

$$\mathcal{L}_i^z(\mathbf{w}_z, b, \mathbf{z}_i, \Sigma_i^z, y_i) \triangleq \frac{1 - y_i(\mathbf{w}_z \cdot \mathbf{z}_i + b)}{2} \left[y_i \operatorname{erf}\left(\frac{y_i - (\mathbf{w}_z \cdot \mathbf{z}_i + b)}{\sqrt{2 \mathbf{w}_z^\top \Sigma_i^z \mathbf{w}_z}} \right) + 1 \right] + \frac{\sqrt{\mathbf{w}_z^\top \Sigma_i^z \mathbf{w}_z}}{\sqrt{2\pi}} \exp\left(-\frac{[y_i - (\mathbf{w}_z \cdot \mathbf{z}_i + b)]^2}{2 \mathbf{w}_z^\top \Sigma_i^z \mathbf{w}_z} \right). \tag{17}$$

Therefore, the objective function $\mathcal{J}' \colon \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$, given by (7) can be approximated as follows

$$\mathcal{J}'(\mathbf{w}, b) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{l} \mathcal{L}_i^z(P_i \mathbf{w}, b, \mathbf{z}_i, \Sigma_i^z, y_i). \tag{18}$$

Following similar arguments as in the case of learning in the original space, $\mathcal{J}'$ can be shown to be convex (see Appendix B).

The first derivative of J' with respect to $\mathbf{w}$ is given as follows

$$\frac{\partial}{\partial \mathbf{w}} \mathcal{J}'(\mathbf{w}, b) = \mathbf{w} + C \sum_{i=1}^{l} \frac{\partial}{\partial \mathbf{w}} \mathcal{L}_i^z(\mathbf{w}_z, b, \mathbf{z}_i, \Sigma_i^z, y_i). \tag{19}$$

Thus, by using the chain rule,

$$\frac{\partial}{\partial \mathbf{w}} \mathcal{J}'(\mathbf{w}, b) = \mathbf{w} + C \sum_{i=1}^{l} \frac{\partial}{\partial \mathbf{w}_z} \mathcal{L}_i^z(\mathbf{w}_z, b, \mathbf{z}_i, \Sigma_i^z, y_i) \frac{\partial \mathbf{w}_z}{\partial \mathbf{w}}, \tag{20}$$

where

$$\frac{\partial \mathbf{w}_z}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}} P_i \mathbf{w} = P_i.$$

By differentiating $\mathcal{L}_i^z$ with respect to $\mathbf{w}_z$, (20) yields

$$\frac{\partial}{\partial \mathbf{w}} \mathcal{J}'(\mathbf{w}, b) = \mathbf{w} + C \sum_{i=1}^{l} \left[\frac{\exp\left(-\frac{[y_i - (\mathbf{w}_z \cdot \mathbf{z}_i + b)]^2}{2 \mathbf{w}_z^\top \Sigma_i^z \mathbf{w}_z} \right)}{\sqrt{2\pi \mathbf{w}_z^\top \Sigma_i^z \mathbf{w}_z}} P_i^\top \big(\Sigma_i^z \mathbf{w}_z\big) - \frac{1}{2} \left(\operatorname{erf}\left(\frac{y_i - (\mathbf{w}_z \cdot \mathbf{z}_i + b)}{\sqrt{2 \mathbf{w}_z^\top \Sigma_i^z \mathbf{w}_z}} \right) + y_i \right) P_i^\top \mathbf{z}_i \right]. \tag{21}$$

Moreover, the first derivative of J' with respect to b is given as follows

$$\frac{\partial}{\partial b} \mathcal{J}'(\mathbf{w}, b) = -\frac{C}{2} \sum_{i=1}^{l} \left[\operatorname{erf}\left(\frac{y_i - (\mathbf{w}_z \cdot \mathbf{z}_i + b)}{\sqrt{2 \mathbf{w}_z^\top \Sigma_i^z \mathbf{w}_z}} \right) + y_i \right], \tag{22}$$

where $\mathbf{w}_z = P_i \mathbf{w}$, $\Sigma_i^z = P_i \Sigma_i P_i^\top$.

At the implementation level, for solving the SVM-GSU in linear subspaces, the eigenanalysis of the covariance matrices Σ_i is performed only once per each Gaussian distribution, before the optimization procedure begins. Consequently, the following orthonormal matrices and vectors are computed once:

- $U_i = [\mathbf{u}_1^i \ \mathbf{u}_2^i \ \dots \ \mathbf{u}_n^i] \in \mathbb{R}^{n \times n}$,
- $U_i' = [\mathbf{u}_1^i \ \mathbf{u}_2^i \ \dots \ \mathbf{u}_{d_i}^i] \in \mathbb{R}^{n \times d_i}$,
- $P_i = U_i'^{\top} \in \mathbb{R}^{d_i \times n}$,
- $\mathbf{z}_i = P_i \mathbf{x}_i$, and
- $\Sigma_i^z = P_i \Sigma_i P_i^{\top} = \Lambda_i^z \in \mathbb{S}_{++}^{d_i}$,

Then, for each iteration of $(\mathbf{w}, b) \in \mathbb{R}^{n+1}$, and for each training example (distribution), $(\mathbf{x}_i, \Sigma_i)$, the projected (normal to the separating hyperplane) vector has to be computed:

$$\mathbf{w}_z = P_i \mathbf{w} \in \mathbb{R}^{d_i}.$$

Finally, the loss function is computed in the low-dimensional spaces $\mathbb{R}^{d_i}$, $i = 1, \dots, l$ as shown in (17). The objective function is computed as shown in (18), while its first derivatives are computed as in (21), (22).

IV. Experiments

The classification performance of the proposed algorithm is initially validated on 2D synthetic data, in order to illustrate how the linear SVM-GSU classifier works. To this end, we consider binary classification toy experiments and validate on them the proposed learning algorithm both in the original feature space, as well as in linear subspaces.

Next, the proposed algorithm is applied to two different, challenging learning problems, i.e., the problem of complex event detection in video, and the problem of image classification of handwritten digits. The large video dataset of the TRECVID Multimedia Event Detection (MED) 2014 task is used for the event detection experiments (Sect. IV-B), while the well-known MNIST database of handwritten digits is used for the image classification ones (Sect. IV-C). For each of those problem domains, a methodology for modeling the uncertainty of each input (random) vector is also proposed.

A. Toy examples using synthetic data

In this subsection, we present two toy examples that provide insights into understanding the way the proposed algorithm works. As shown in Fig.2, two toy artificial binary classification problems are constructed. Negative samples are denoted by red $\times$ marks, while positive ones by green crosses. We assume that the uncertainty of each training example is given via a covariance matrix. In Fig.2a and 2c, the ellipses show the iso-density loci of points described by the 0.03% of the maximum density of each Gaussian distribution (please note that these ellipses are only used for visualization purposes). Moreover, in Fig.2b and 2d, the covariance matrices are approximated by low-rank matrices (rank one).

For each of the above experiments, a linear baseline SVM (LSVM) is trained using solely the centres of the distributions; i.e., ignoring the uncertainty of each sample. The resulting separating lines are shown in Fig.2 in dashed red. Moreover, a linear SVM-GSU (LSVM-GSU) is also trained using the centres of the above distributions, and the covariance matrices; i.e., using the parameters of the Gaussian distribution followed by each training example. LSVM-GSU is trained first in the original feature space ($\mathbb{R}^2$), and then in linear subspaces ($\mathbb{R}$), preserving for each covariance matrix 90% of the total variance. The resulting separating lines virtually coincide and are shown in Fig.2a and 2c (solid green lines). Finally, the resulting separating lines of the SVM-GSUs trained in linear subspaces using the low-rank (rank one) covariance matrices and preserving 90% of the total variance are shown with green lines in Fig.2b and 2d. It is evident that, when the uncertainty of the training data is taken into consideration, the decision boundaries may change drastically. Finally, the proposed algorithm achieves to learn approximately the same (or a very similar) separating line, even in the cases where the optimization problem is approximated in linear subspaces, or the covariance matrices of the input vectors are low-rank.

B. Video Event Detection

1) Dataset and experimental setup: For experiments on video event detection, the large-scale video dataset of the TRECVID Multimedia Event Detection (MED) 2014 task [28] is used. The ground-truth annotated portion of it consists of three different video subsets: the "pre-specified" (PS) video subset (2000 videos, 80 hours, 20 event classes), the "ad-hoc" (AH) video subset (1000 videos, 40 hours, 10 event classes), and the "background" (BG) video subset (5000 videos, 200 hours). Each video in the above dataset belongs to either one of 30 target event classes, or to the "rest of the world" (background) class. The above video dataset (PS+AH+BG) is partitioned such that a training and an evaluation set are created, as follows

- **Training Set**
  - 50 positive samples per event class,
  - 2496 background samples (negative for all event classes).
- **Evaluation Set**
  - ~ 50 positive samples per event class,
  - 2496 background samples (negative for all event classes).

A model vector representation scheme is adopted, similarly to [29], for representing videos. That is, a set of 346 pre-existing visual concept detectors (linear SVM classifiers that are trained on the TRECVID Semantic Indexing (SIN) 2014 dataset [29], [28]) is used for deriving a 346-element descriptor vector for each video (hereafter called "model vector"). Specifically, each input video stream is initially sampled such that a keyframe is generated every 6 seconds. Next, each keyframe is processed as discussed above and a keyframe-level model vector is computed. Then, a video-level model vector for each video is computed by taking the average of the corresponding keyframe-level representations. Thus, the keyframe-level model vectors can be seen as different observations of the model vector which represents each video.

2) Uncertainty modeling: Let us now define a set $\mathcal{X}$ of l annotated random vectors representing the aforementioned video-level model vectors. Each random vector is distributed

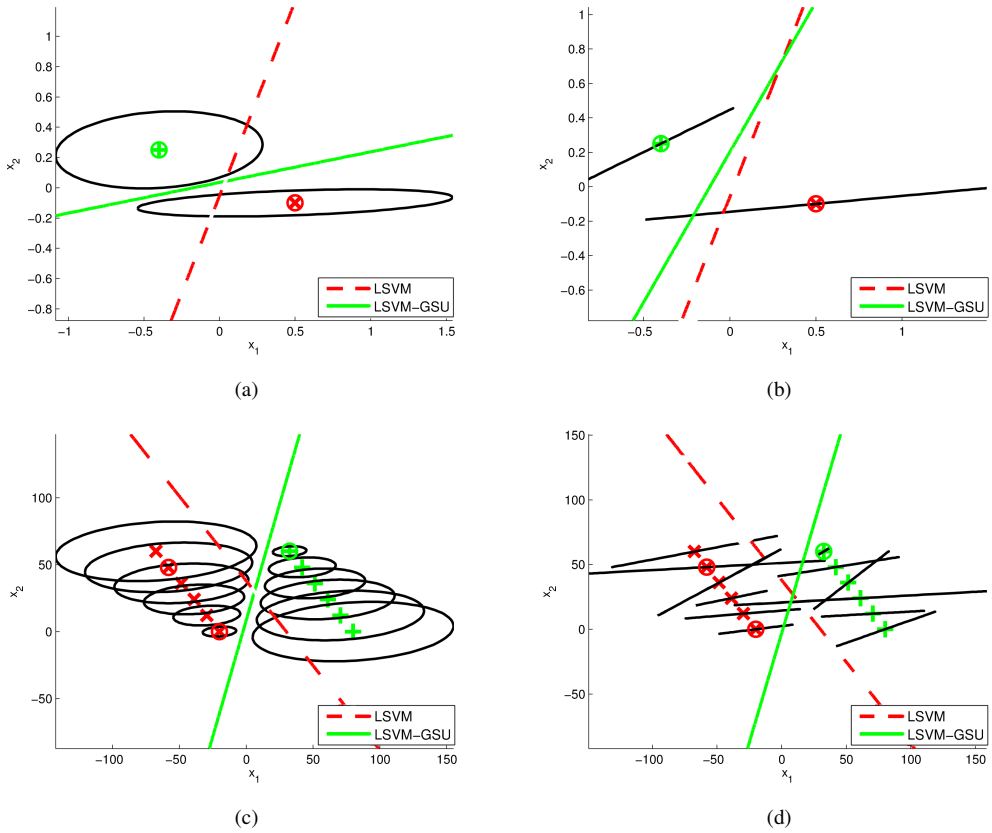


Fig. 2: Toy experiments illustrating LSVM-GSU (green solid line) learning in the original feature space (a,c), and in linear subspaces (b,d), in comparison with the baseline LSVM (red dashed lines). Circled points indicate support vectors as identified by the standard LSVM.

normally; i.e., for the random vector representing the i-th video, $\mathbf{X}_i$, we have $\mathbf{X}_i \sim \mathcal{N}(\mathbf{x}_i, \Sigma_i)$. That is, $\mathcal{X} = \{(\mathbf{x}_i, \Sigma_i, y_i) \colon \mathbf{x}_i \in \mathbb{R}^n, \Sigma_i \in \mathbb{S}^n_{++}, y_i \in \{\pm 1\},\ i = 1, \ldots, l\}$. For each random vector $\mathbf{X}_i$, a number, N_i, of observations, $\{\mathbf{x}_i^t \in \mathbb{R}^n \colon t = 1, \ldots, N_i\}$ is available (these are the keyframe-level model vectors that have been computed). Then, the mean vector and the covariance matrix of $\mathbf{X}_i$ are computed respectively as follows

$$\mathbf{x}_i = \frac{1}{N_i} \sum_{t=1}^{N_i} \mathbf{x}_i^t, \tag{23}$$

$$\Sigma_i = \sum_{t=1}^{N_i} (\mathbf{x}_i^t - \mathbf{x}_i)(\mathbf{x}_i^t - \mathbf{x}_i)^\top. \tag{24}$$

However, the number of observations per each video that are available for our dataset is in most cases much lower than the dimensionality of the input space; for instance, the average number of observations available for each random vector (video-level representation) is approximately 20 model vectors (keyframe-level representations), while the dimensionality of the input space is $n = 346$. Consequently, the covariance matrices that arise using (24) are typically low-rank; i.e. $\mathrm{rank}(\Sigma_i) \leq N_i$. To overcome this issue, we assume that the desired covariance matrices are diagonal. That is, we require that the covariance matrix of the i-th training sample is given by

$$\widehat{\Sigma_i} = \mathrm{diag}\left(\hat{\sigma}_i^1, \ldots, \hat{\sigma}_i^n\right),$$

such that the squared Frobenious norm of the difference $\Sigma_i - \widehat{\Sigma_i}$ is minimized, i.e.,

$$\widehat{\Sigma_i} = \underset{\hat{\sigma}_i^t | t=1,\ldots,n}{\mathrm{argmin}} \left\| \Sigma_i - \mathrm{diag}\left(\hat{\sigma}_i^1, \ldots, \hat{\sigma}_i^n\right) \right\|_F^2.$$

It can easily be shown that the above criterion is fulfilled when the estimator covariance matrix $\widehat{\Sigma_i}$ is equal to the diagonal part of the sample covariance matrix Σ_i, i.e.

$$\widehat{\Sigma_i} = \mathrm{diag}\left(\sigma_i^1, \ldots, \sigma_i^n\right).$$

We note that, using this approximation approach, the covariance matrices are diagonal but anisotropic and different for each training input example. This is in contrast with other methods (e.g. [7], [17]) that assume more restrictive modeling approaches for the uncertainty; i.e., isotropic noise for each training sample.

3) Experimental results: Table I shows the performance of the proposed linear SVM-GSU (LSVM-GSU) in terms of average precision (AP) [30] for each target event in comparison with the baseline linear SVM (LSVM), as well as with a linear SVM extension which handles the input uncertainty isotropically (LSVM-isotropic) as in [7], [17]. Moreover, for each dataset, the mean average precision (MAP) across all

target events is reported. The optimization of the C parameter for both LSVM and LSVM-GSU is performed using a line search on a 3-fold cross-validation procedure, where at each fold the training set is split to 70% learning set and 30% validation set.

In Table I, column (a) shows the performance of the baseline LSVM when training is carried out using keyframe-level model vectors. That is, in this experimental scenario we attempt to resemble the case where a standard LSVM is trained using all the available observations of each training distribution, in contrast with the proposed LSVM-GSU, where training is carried out using solely the mean vectors and the covariance matrices. In column (b), we report the results of the standard LSVMs which were trained using the video-level representations; that is, solely the mean vectors of each distribution. In contrast, by modeling the uncertainty as described in the previous section, the proposed LSVM-GSU is validated both in the case that learning is carried out in the original feature space (column (h)), and in the cases that it is approximated in linear subspaces by preserving a certain fraction (p) of the total variance of each covariance matrix. Columns (d)-(g) show the performance of LSVM-GSU when $p = 0.75, 0.90, 0.95$, and 0.99, respectively. The performance of the SVM extension, described in [7], [17], where uncertainty is modeled isotropically (LSVM-isotropic) is given in column (c). The bold-faced numbers indicate the best result achieved for each event class. Finally, in column (i), the results of the McNemar [31], [32], [33] statistical significance test are reported. A $*$ denotes statistically significant differences between the proposed LSVM-GSU (learning in original space) and baseline LSVM, while a $\sim$ denotes statistically significant differences between LSVM-GSU and LSVM-isotropic.

From the obtained results, we observe that the proposed algorithm (learning in the original feature space) achieved better detection performance than both LSVM and LSVM-isotropic for 22 out of the 30 event classes. The relative boost between LSVM-GSU and LSVM, achieved for each event class, is shown in column (j) of Table I, while the overall best relative performance boost (in MAP) is equal to 9.83% and is achieved when LSVM-GSU is learned in the original feature space. However, it is worth noting that a considerable boost was also achieved when the LSVM-GSU is approximated in linear subspaces by preserving the 99% of the total variance for each covariance matrix. Furthermore, in general we observe that, as the fraction of the total variance preserved decreases, the overall detection performance also decreases.

C. Hand-written digit classification

1) Dataset and experimental setup: The proposed algorithm is also validated on the problem of image classification using the MNIST dataset of handwritten digits [34]. The MNIST dataset provides a training set of 60000 samples (approx. 6000 samples per digit), and a test set of 10000 samples (approx. 1000 samples per digit). Each sample is represented by a 28×28 8-bit image. Originally, MNIST does not provide any information about the uncertainty of each image; some typical examples of the original training and testing set images are shown in Fig.3a.

In order to make the dataset more challenging, as well as to model a realistic distortion that may happen to this kind of images (scanned handwritten digits), the original MNIST dataset was "polluted" with noise. More specifically, each image example was rotated by a random angle uniformly drawn from the range $[-\theta, +\theta]$, where θ is measured in degrees. Moreover, each image was translated by a random vector $\mathbf{t}$ uniformly drawn from $[-t_p, +t_p]^2$, where t_p is a positive integer expressing distance that is measured in pixels. We created five different noisy datasets by setting $\theta = 15^\circ$ and $t_p \in \{3, 5, 7, 9, 11\}$. The polluted datasets (D_1 to D_5, respectively) are shown in Table II, where D_0 denotes the original MNIST dataset. Fig. 3b and 3c show illustrative examples of the noisy datasets D_2 ($\theta = 15^\circ$, $t_p = 5$) and D_5 ($\theta = 15^\circ$, $t_p = 11$), respectively. Experiments with θ in range $[5^\circ, 25^\circ]$ gave very similar results, thus we chose to solely report the results that correspond to $\theta = 15^\circ$.

TABLE II: MNIST "1" versus "7" datasets

Dataset	θ	t_p
D_0	0°	0
D_1	15°	3
D_2	15°	5
D_3	15°	7
D_4	15°	9
D_5	15°	11

We create six different experimental scenarios using the above datasets (D_0-D_5). First, we defined the problem of discriminating the number one ("1") from the number seven ("7") similarly to [35]. Each class in the training procedure consists of 25 samples, randomly chosen from the pool of digits one ($\sim$ 6K totally) and seven ($\sim$ 6K totally), while the evaluation of the trained classifier is carried out on the full testing set ($\sim$ 2K samples). In each experimental scenario we report the average of 100 runs. Moreover, in each experimental scenario we compare the proposed linear SVM-GSU (LSVM-GSU) to the baseline linear SVM (LSVM), as well as to LSVM-isotropic ([7], [17]). We report the average precision (AP) [30] for each target class, and the mean average precision (MAP) across 100 runs.

2) Uncertainty modeling: In Appendix C, we propose a methodology that, given an image, models the distribution of the image that results by a random translation of it. The methodology is a first-order Taylor approximation, in a way similar to one used for optical flow. Then, we can show that the image representation is distributed normally with a certain mean vector and covariance matrix, which are also being evaluated. We use this methodology for modeling the uncertainty of each training image in all the experiments below. More specifically, we assume that the translation is distributed normally as $\mathbf{t} \sim \mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)$, where

$$\boldsymbol{\mu}_t = (0, 0)^\top,$$

and

$$\Sigma_t = \begin{pmatrix} \sigma_h^2 & 0 \\ 0 & \sigma_v^2 \end{pmatrix}.$$

TABLE I: Event detection performance (AP and MAP) of the linear SVM-GSU compared to the baseline linear SVM and a LSVM extension for handling isotropic uncertainty (as in [7], [17]) using the MED14 dataset.

Event Class	**LSVM (AP)**		**(c) LSVM isotropic (AP)**	**LSVM-GSU**						
				Learning in linear subspaces (AP)				**Learning in the original space**		
	(a) keyframe level	**(b) video level**		**(d)** $p = 0.75$	**(e)** $p = 0.90$	**(f)** $p = 0.95$	**(g)** $p = 0.99$	**(h) AP**	**(i) McNemar tests**	**(j) Boost (%)**
E021	0.1130	0.1862	**0.2018**	0.1156	0.1073	0.1200	0.1565	0.1994	$*, \sim$	7.09
E022	0.1244	0.1262	0.1492	0.0863	0.1107	0.0971	**0.1610**	0.1583		25.44
E023	0.2680	0.2593	0.2647	0.1570	0.2585	0.2432	0.2452	**0.2733**	$*$	5.40
E024	0.0467	0.0500	0.0540	0.0476	0.0537	0.0452	0.0492	**0.0596**	$*, \sim$	19.20
E025	**0.0252**	0.0169	0.0077	0.0184	0.0195	0.0173	0.0195	0.0077		-54.44
E026	0.0750	0.0700	0.0681	0.0733	**0.0872**	0.0851	0.0707	0.0810	$*, \sim$	15.71
E027	0.2502	0.2666	0.2504	0.1665	0.2344	0.2799	**0.3105**	0.2914	$*, \sim$	9.30
E028	0.1948	0.1829	0.1983	0.1693	0.2007	**0.2091**	0.2027	0.2064		12.85
E029	**0.2458**	0.2330	0.2433	0.2319	0.2299	0.2520	0.2250	0.2337		0.30
E030	0.1054	0.0601	0.1034	0.0755	0.0842	0.0914	0.1100	**0.1179**	$*$	96.17
E031	0.1781	0.1992	0.2133	0.1105	0.1603	**0.2422**	0.2291	0.2125	$*, \sim$	6.68
E032	0.0653	0.0521	0.0613	0.0484	0.0599	**0.0673**	0.0654	0.0638	$\sim$	22.46
E033	0.1019	0.0935	0.1335	0.1162	**0.1497**	0.1287	0.1363	0.1370	$*, \sim$	46.52
E034	0.0711	0.0658	0.0725	0.0692	**0.0728**	0.0719	0.0707	0.0726	$*, \sim$	10.33
E035	0.1996	0.2648	**0.2794**	0.1476	0.1651	0.1812	0.2207	0.2742	$*, \sim$	3.55
E036	0.1674	0.1957	0.2141	0.2191	0.2209	0.2281	0.2235	**0.2436**	$*, \sim$	24.48
E037	0.2227	0.3742	0.3728	0.3246	0.3606	0.3894	**0.3913**	0.3595	$*, \sim$	-3.93
E038	0.0567	**0.0791**	0.0360	0.0719	0.0692	0.0732	0.0680	0.0757	$*, \sim$	-4.30
E039	0.2189	0.2419	0.2397	0.1668	0.1645	0.1953	0.2210	**0.2454**	$*, \sim$	1.45
E040	0.0957	0.0829	0.1197	0.1251	0.1346	**0.1484**	0.1444	0.1281		54.52
E041	0.0656	0.0835	0.0890	0.0637	0.0653	0.0812	0.0839	**0.0941**	$*, \sim$	12.69
E042	0.0622	0.0580	0.0681	**0.0757**	0.0721	0.0726	0.0701	0.0753		29.83
E043	**0.2212**	0.2063	0.1996	0.1321	0.1650	0.2160	0.2055	0.1984	$*, \sim$	-3.83
E044	0.1631	0.2844	0.2999	0.1467	0.1828	0.2547	0.3008	**0.3090**		8.65
E045	0.1348	0.1773	0.1723	0.1249	0.1557	**0.1948**	0.1804	0.1853	$*, \sim$	4.51
E046	0.0750	0.0814	0.0862	0.0713	0.0626	0.0712	**0.1032**	0.1017	$*, \sim$	24.94
E047	0.1208	0.1275	**0.1329**	0.1213	0.1240	0.1304	0.1298	0.1316	$*$	3.22
E048	0.0476	0.0613	0.0772	0.0530	**0.0941**	0.0667	0.0570	0.0673	$*, \sim$	9.79
E049	0.0658	0.1067	**0.1431**	0.0458	0.0327	0.0494	0.1082	0.1184		10.97
E050	0.1849	0.2226	0.2256	0.1982	0.2522	**0.2604**	0.2447	0.2306	$*, \sim$	3.59
MAP	0.1322	0.1503	0.1592	0.1191	0.1383	0.1521	0.1601	**0.1651**		9.83

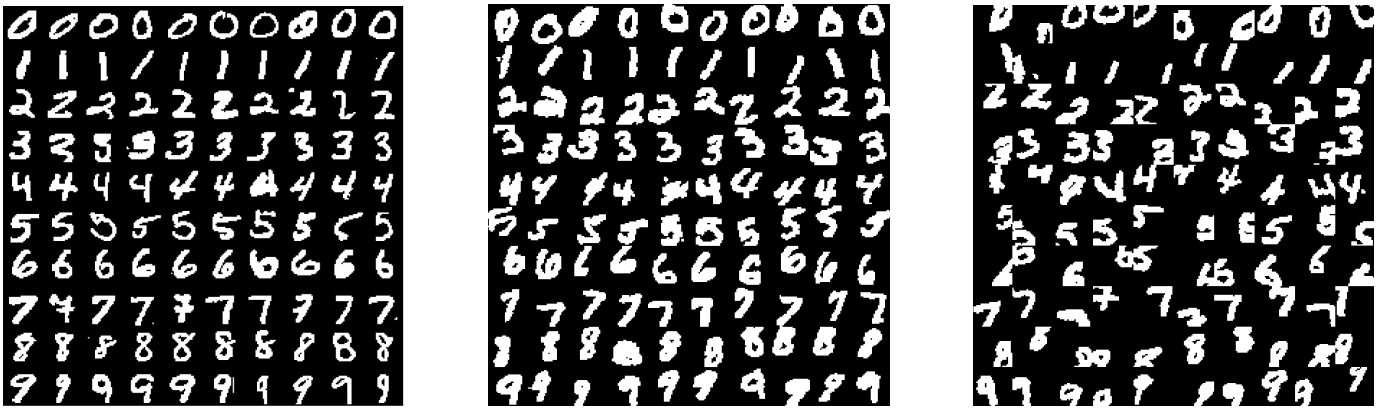
The variances of the horizontal and the vertical components of the translation, namely σ_h^2 and σ_v^2, are set to

$$\sigma_h^2 = \sigma_v^2 = \left(\frac{p_t}{3}\right)^2,$$

where p_t is measured in pixels. That is, the covariance matrix is set such that the translation falls in the square $[-p_t, p_t] \times [-p_t, p_t]$ with probability 99.7%. For the experiments described below, this parameter is set to $p_t = 5$ pixels. Using the above, the mean vector and covariance matrix of the i-th image are given by (34) and (35), respectively, in Appendix C.

3) Experimental results: Table III shows the performance of the proposed classifier (LSVM-GSU) in terms of mean average precision (MAP) for the problem of discriminating digit "1" to "7", for each dataset defined above (D_0-D_5). We report the average of 100 runs of each experiment. The proposed algorithm is compared both to the baseline linear SVM (LSVM), where the uncertainty of each training sample is not taken into account, as well as to a linear SVM extension where the uncertainty is taken into consideration isotropically (LSVM-isotropic) as in [7], [17]. The optimization of the C parameter for both LSVM and LSVM-GSU is performed using a line search on a 3-fold cross-validation procedure, where at each fold the training set is split to 70% learning set and 30% validation set. The performance of LSVM-GSU when the training of each classifier is carried out in the original feature space is shown in row 4, and in linear subspaces in row 5. In row 5 we report both the classification performance, and in parentheses the fraction of variance that resulted in the best classification result.

The performance of the baseline linear SVM is shown in the second row, and the performance of the linear SVM extension handling the noise isotropically (as in [7], [17]) is shown in the third row. Moreover, Fig. 4 shows the results of the above experimental scenarios for datasets D_0-D_5. The horizontal axis of each subfigure describes the fraction of the total variance preserved for each covariance matrix (p), while the vertical axis shows the respective performance of LSVM-GSU with learning in linear subspaces (LSVM-GSU-SL$_p$). Furthermore, in each subfigure, for $p = 1$ we also draw the result of the proposed LSVM-GSU in the original feature space (denoted with a rhombus), as well as the result of the linear SVM extension that handles the uncertainty isotropically (LSVM-isotropic) [7], [17] (denoted with a star). We report the mean, and with an errorbar show the variance of the 100 iterations. The performance of the baseline LSVM is shown with a solid line, while two dashed lines show



(a) (b) (c)

Fig. 3: The MNIST dataset of handwritten digits. Illustrative examples of: (a) the original dataset, and the generated noisy datasets (b) D_2 ($\theta = 15°$, $t_p = 5$) and (c) D_5 ($\theta = 15°$, $t_p = 11$).

the corresponding variance of the 100 runs. From the obtained results, we observe that the proposed LSVM-GSU with learning in linear subspaces outperforms both the baseline LSVM and LSVM-isotropic for all datasets D_0-D_5. Moreover, LSVM-GSU achieves better classification results than LSVM-isotropic in 5 out of 6 datasets, when learning is carried out in the original feature space. Finally, all the reported results are shown to be statistically significant using the t-test [36]; significance values (p-values) were much lower than the significance level of 1%, with most values being near 10^{-4}.

V. Conclusion

In this paper we proposed a novel classifier that efficiently exploits uncertainty in its input under the SVM paradigm. The proposed SVM-GSU was validated on the large-scale dataset of TRECVID MED 2014 for the problem of video event detection, as well as on the MNIST dataset of handwritten digits. For both of the above problems, a method for modeling and estimating the uncertainty of each training example was also proposed. As a shown by the experiments, SVM-GSU, validated in the video event detection and the image classification problems, efficiently takes into consideration the uncertainty of the training examples and achieves better detection or classification performance than the standard SVM, and previous SVM extensions that model uncertainty isotropically.

Appendix A
On Gaussian-like Integrals over Half-spaces

Theorem 1. *Let $\mathbf{X} \in \mathbb{R}^n$ be a random vector that follows a multivariate Gaussian distribution with mean vector $\boldsymbol{\mu} \in \mathbb{R}^n$ and covariance matrix $\Sigma \in \mathbb{S}^n_{++}$, where $\mathbb{S}^n_{++}$ denotes the space of $n \times n$ symmetric positive definite matrices with real entries. The probability density function (pdf) of $\mathbf{X}$ is given by $f_{\mathbf{X}} \colon \mathbb{R}^n \to \mathbb{R}$,*

$$f_{\mathbf{X}}(\mathbf{x}; \boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right).$$

Moreover, let $\mathcal{H}$ be the hyperplane given by $\mathbf{a} \cdot \mathbf{x} + b = 0$. $\mathcal{H}$ divides the Euclidean n-dimensional space into two half-spaces (an open and a closed one), where the closed upper half-space is given by

$$\Omega_+ = \{\mathbf{x} \in \mathbb{R}^n \colon \mathbf{a} \cdot \mathbf{x} + b \geq 0\}.$$

Then, the function $I_+ \colon (\mathbb{R}^n \times \mathbb{R}) \times (\mathbb{R}^n \times \mathbb{S}^n_{++}) \to \mathbb{R}$, defined as

$$I_+(\mathbf{a}, b, \boldsymbol{\mu}, \Sigma) \triangleq \int_{\Omega_+} (\mathbf{a} \cdot \mathbf{x} + b) f_X(\mathbf{x}) \, \mathrm{d}\mathbf{x}, \tag{25}$$

is equal to

$$\frac{\mathbf{a} \cdot \boldsymbol{\mu} + b}{2} \left[1 + \operatorname{erf}\left(\frac{\mathbf{a} \cdot \boldsymbol{\mu} + b}{\sqrt{2\mathbf{a}^\top \Sigma \mathbf{a}}}\right)\right] + \frac{\sqrt{\mathbf{a}^\top \Sigma \mathbf{a}}}{\sqrt{2\pi}} \exp\left(-\left(\frac{\mathbf{a} \cdot \boldsymbol{\mu} + b}{\sqrt{2\mathbf{a}^\top \Sigma \mathbf{a}}}\right)^2\right), \tag{26}$$

where $\operatorname{erf} \colon \mathbb{R} \to (-1, 1)$, $x \mapsto \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} \, \mathrm{d}t$ is the so-called error function. Moreover, if the half-space is given as the lower half-space $\Omega_- = \{\mathbf{x} \in \mathbb{R}^n \colon \mathbf{a} \cdot \mathbf{x} + b \leq 0\}$, then the function $I_- \colon (\mathbb{R}^n \times \mathbb{R}) \times (\mathbb{R}^n \times \mathbb{S}^n_{++}) \to \mathbb{R}$, given by

$$I_+(\mathbf{a}, b, \boldsymbol{\mu}, \Sigma) \triangleq \int_{\Omega_-} (\mathbf{a} \cdot \mathbf{x} + b) f_X(\mathbf{x}) \, \mathrm{d}\mathbf{x}, \tag{27}$$

is equal to

$$\frac{\mathbf{a} \cdot \boldsymbol{\mu} + b}{2} \left[1 - \operatorname{erf}\left(\frac{\mathbf{a} \cdot \boldsymbol{\mu} + b}{\sqrt{2\mathbf{a}^\top \Sigma \mathbf{a}}}\right)\right] - \frac{\sqrt{\mathbf{a}^\top \Sigma \mathbf{a}}}{\sqrt{2\pi}} \exp\left(-\left(\frac{\mathbf{a} \cdot \boldsymbol{\mu} + b}{\sqrt{2\mathbf{a}^\top \Sigma \mathbf{a}}}\right)^2\right). \tag{28}$$

Proof: We begin with the integral in (25). In our approach we will need several coordinate transforms. First, we start with a translation in order to get rid of the mean:

$$\mathbf{y} = \mathbf{x} - \boldsymbol{\mu} \Leftrightarrow \mathbf{x} = \mathbf{y} + \boldsymbol{\mu}.$$

Then

$$I_+(\mathbf{a}, b, \boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \int_{\Omega_1^+} (\mathbf{a} \cdot \mathbf{y} + \mathbf{a} \cdot \boldsymbol{\mu} + b) \exp\left(-\frac{1}{2}\mathbf{y}^\top \Sigma^{-1} \mathbf{y})\right) \mathrm{d}\mathbf{y},$$

TABLE III: MNIST "1" versus "7" experimental results (MAP). The proposed LSVM-GSU is compared to the baseline linear SVM (LSVM), and a linear SVM extension which handles the uncertainty isotropically (LSVM-isotropic), as in [7], [17].

Dataset		D_0	D_1	D_2	D_3	D_4	D_5
LSVM		0.9952	0.9362	0.8240	0.6830	0.6558	0.6027
LSVM-isotropic		0.9968	0.9327	0.8133	0.7222	0.6675	0.6328
LSVM-GSU	Learning in original space	0.9971	0.9452	0.8310	0.7216	0.6708	0.6353
	Learning in linear subspaces	**0.9972** (0.99)	**0.9480** (0.97)	**0.8562** (0.89)	**0.7543** (0.85)	**0.6974** (0.95)	**0.6640** (0.25)

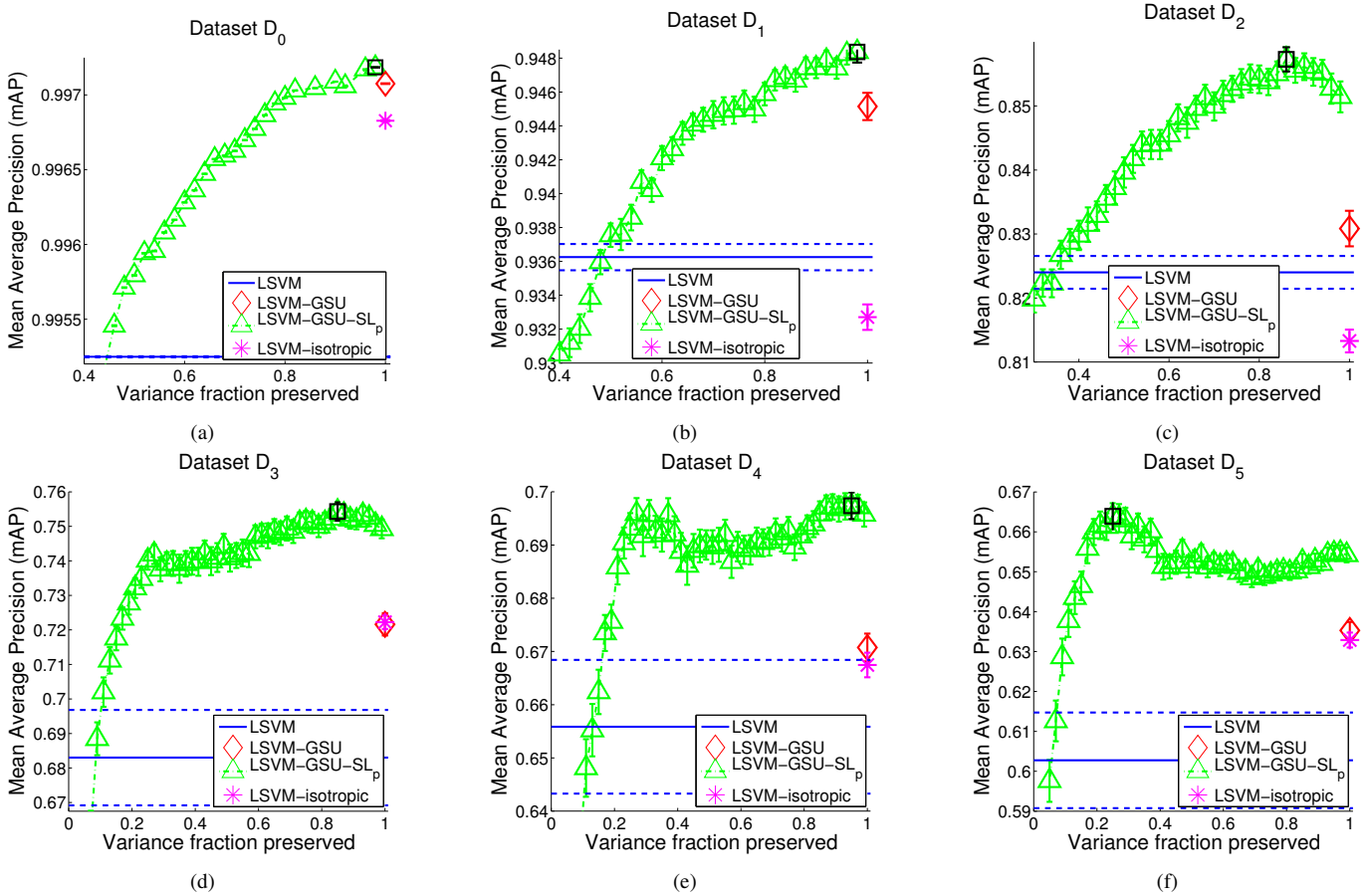


Fig. 4: Comparisons between the proposed LSVM-GSU, the baseline LSVM, and the LSVM with isotropic noise in (a) the original MNIST dataset (D_0), and (b)-(f) the noisy generated datasets D_1-D_5.

where

$$\Omega_1^+ = \{\mathbf{y} \in \mathbb{R}^n \colon \mathbf{a} \cdot \mathbf{y} + \mathbf{a} \cdot \boldsymbol{\mu} + b \geq 0\}.$$

Next, since $\Sigma \in \mathbb{S}_{++}^n$, there exist an orthonormal matrix U and a diagonal matrix D with positive elements, i.e. the eigenvalues of Σ, such that $\Sigma = U^\top D U$. Thus, it holds that $\Sigma^{-1} = (U^\top D U)^{-1} = U^{-1} D^{-1} (U^\top)^{-1} = U^\top D^{-1} U$. Then, by letting $\mathbf{z} = U\mathbf{y}$ and $\mathbf{a}_1 = U\mathbf{a}$, we have

$$\mathbf{a} \cdot \mathbf{y} = \mathbf{a}^\top \mathbf{y} = \mathbf{a}^\top (U^{-1} U)\mathbf{y} = \mathbf{a}^\top U^\top U \mathbf{z} = \mathbf{a}_1^\top \mathbf{z},$$

and

$$\mathbf{y}^\top \Sigma^{-1} \mathbf{y} = \mathbf{y}^\top (U^\top D U)^{-1} \mathbf{y} = (\mathbf{y}^\top U^\top) D^{-1} (U\mathbf{y}) = (U\mathbf{y})^\top D^{-1} (U\mathbf{y}) = \mathbf{z}^\top D^{-1} \mathbf{z}.$$

Then

$$I_+(\mathbf{a}, b, \boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \int_{\Omega_2^+} (\mathbf{a}_1 \cdot \mathbf{z} + \mathbf{a} \cdot \boldsymbol{\mu} + b) \exp\left(-\frac{1}{2} \mathbf{z}^\top D^{-1} \mathbf{z}\right) \mathrm{d}\mathbf{z},$$

where

$$\Omega_2^+ = \{\mathbf{z} \in \mathbb{R}^n \colon \mathbf{a}_1 \cdot \mathbf{z} + \mathbf{a} \cdot \boldsymbol{\mu} + b \geq 0\},$$

since for the Jacobian $J = |U|$, it holds that $|J| = 1$.

Now, in order to do rescaling, we set $\mathbf{z} = D^{\frac{1}{2}} \mathbf{v}$ and $\mathbf{a}_2 = D^{\frac{1}{2}} \mathbf{a}_1$. Thus,

$$\mathbf{z}^\top D^{-1} \mathbf{z} = (D^{\frac{1}{2}} \mathbf{v})^\top D^{-1} (D^{\frac{1}{2}} \mathbf{v}) = \mathbf{v}^\top (D^{\frac{1}{2}} D^{-1} D^{\frac{1}{2}}) \mathbf{v} = \mathbf{v}^\top \mathbf{v}.$$

Moreover, $\mathbf{a}_1^\top \mathbf{z} = \mathbf{a}_1^\top (D^{\frac{1}{2}} \mathbf{v}) = (D^{\frac{1}{2}} \mathbf{a}_1)^\top \mathbf{v} = \mathbf{a}_2^\top \mathbf{v}$. Also, it holds that $D^{\frac{1}{2}} = |\Sigma|^{\frac{1}{2}}$ and $\mathrm{d}\mathbf{z} = |D^{\frac{1}{2}}| \mathrm{d}\mathbf{v} = |\Sigma|^{\frac{1}{2}} \mathrm{d}\mathbf{v}$.

Consequently,

$$I_+(\mathbf{a}, b, \boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}}} \int_{\Omega_3^+} (\mathbf{a}_2 \cdot \mathbf{v} + \mathbf{a} \cdot \boldsymbol{\mu} + b) \exp\left(-\frac{1}{2}\mathbf{v}^\top \mathbf{v}\right) \mathrm{d}\mathbf{v},$$

where

$$\Omega_3^+ = \{\mathbf{v} \in \mathbb{R}^n \colon \mathbf{a}_2 \cdot \mathbf{v} + \mathbf{a} \cdot \boldsymbol{\mu} + b \geq 0\}.$$

Now, let B be an orthogonal matrix such that $B\mathbf{a}_2 = \|\mathbf{a}_2\|\mathbf{e}_n$, which also means that $\mathbf{a}_2 = B^{-1}\|\mathbf{a}_2\|\mathbf{e}_n = B^\top \|\mathbf{a}_2\|\mathbf{e}_n$. Moreover, let $\mathbf{m} = B\mathbf{v}$. Then,

$$\mathbf{a}_2 \cdot \mathbf{v} = \mathbf{a}_2^\top \mathbf{v} = (B^\top \|\mathbf{a}_2\|\mathbf{e}_n)^\top \mathbf{v} = \|\mathbf{a}_2\|\mathbf{e}_n^\top (B\mathbf{v}) = \|\mathbf{a}_2\|\mathbf{e}_n^\top \mathbf{m}.$$

Moreover,

$$\mathbf{v}^\top \mathbf{v} = \mathbf{v}^\top (B^{-1}B)\mathbf{v} = \mathbf{v}^\top (B^\top B)\mathbf{v}(B\mathbf{v})^\top (B\mathbf{v}) = \mathbf{m}^\top \mathbf{m}.$$

Then

$$I_+(\mathbf{a}, b, \boldsymbol{\mu}, \Sigma) = \frac{1}{\sqrt{2\pi}} \int_c^{+\infty} (\|\mathbf{a}_2\| t + \mathbf{a} \cdot \boldsymbol{\mu} + b) \exp\left(-\frac{1}{2}t^2\right) \mathrm{d}\mathbf{t},$$

where

$$\Omega_4^+ = \{\mathbf{m} \in \mathbb{R}^n \colon \|\mathbf{a}_2\|\mathbf{e}_n^\top \mathbf{m} + \mathbf{a} \cdot \boldsymbol{\mu} + b \geq 0\} = \mathbb{R}^{n-1} \times [c, +\infty),$$

and $c = -\frac{\mathbf{a} \cdot \boldsymbol{\mu} + b}{\|\mathbf{a}_2\|}$. The norm of $\mathbf{a}_2$ can be expressed in terms of $\mathbf{a}, \Sigma$ as follows

$$\|\mathbf{a}_2\| = \sqrt{(}D)U\mathbf{a} \Longrightarrow \|\mathbf{a}_2\|^2 = (\sqrt{(}D)U\mathbf{a})^\top (\sqrt{(}D)U\mathbf{a}) = \mathbf{a}^\top U^\top \sqrt{(}D)\sqrt{(}D)U\mathbf{a} = \mathbf{a}^\top \Sigma \mathbf{a},$$

and thus

$$I_+(\mathbf{a}, b, \boldsymbol{\mu}, \Sigma) = \frac{1}{\sqrt{2\pi}} \int_c^{+\infty} (\sqrt{\mathbf{a}^\top \Sigma \mathbf{a}}\, t + \mathbf{a} \cdot \boldsymbol{\mu} + b) \exp\left(-\frac{1}{2}t^2\right) \mathrm{d}\mathbf{t}, \quad (29)$$

where $c = -\frac{\mathbf{a} \cdot \boldsymbol{\mu} + b}{\sqrt{\mathbf{a}^\top \Sigma \mathbf{a}}}$, and it is easily evaluated as follows

$$I_+(\mathbf{a}, b, \boldsymbol{\mu}, \Sigma) = \frac{\mathbf{a} \cdot \boldsymbol{\mu} + b}{2}\left[1 + \operatorname{erf}\left(\frac{\mathbf{a} \cdot \boldsymbol{\mu} + b}{\sqrt{2\mathbf{a}^\top \Sigma \mathbf{a}}}\right)\right] + \frac{\sqrt{\mathbf{a}^\top \Sigma \mathbf{a}}}{\sqrt{2\pi}} \exp\left(-\left(\frac{\mathbf{a} \cdot \boldsymbol{\mu} + b}{\sqrt{2\mathbf{a}^\top \Sigma \mathbf{a}}}\right)^2\right).$$

Following similar arguments as above, for $\Omega_- = \{\mathbf{x} \in \mathbb{R}^n \colon \mathbf{a} \cdot \mathbf{x} + b \leq 0\}$, with $\Omega_4^- = \mathbb{R}^{n-1} \times (+\infty, c]$, we have

$$I_-(\mathbf{a}, b, \boldsymbol{\mu}, \Sigma) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{c} (\sqrt{\mathbf{a}^\top \Sigma \mathbf{a}}\, t + \mathbf{a} \cdot \boldsymbol{\mu} + b) \exp\left(-\frac{1}{2}t^2\right) \mathrm{d}\mathbf{t},$$

which leads to

$$I_-(\mathbf{a}, b, \boldsymbol{\mu}, \Sigma) = \frac{\mathbf{a} \cdot \boldsymbol{\mu} + b}{2}\left[1 - \operatorname{erf}\left(\frac{\mathbf{a} \cdot \boldsymbol{\mu} + b}{\sqrt{2\mathbf{a}^\top \Sigma \mathbf{a}}}\right)\right] - \frac{\sqrt{\mathbf{a}^\top \Sigma \mathbf{a}}}{\sqrt{2\pi}} \exp\left(-\left(\frac{\mathbf{a} \cdot \boldsymbol{\mu} + b}{\sqrt{2\mathbf{a}^\top \Sigma \mathbf{a}}}\right)^2\right).$$

■

APPENDIX B
ON THE CONVEXITY OF THE SVM-GSU LOSS FUNCTION

Let $\mathcal{J}$ be the objective function of the optimization problem (4), as shown in (7). We will show that $\mathcal{J}$ is convex with respect to the optimization variables, $\mathbf{w}$ and b, over $\mathbb{R}^n \times \mathbb{R}$. First, as every norm is convex, and every non-negative weighted sum preserves the convexity, it suffices to show that $\mathcal{L}$, as shown in (5), is convex with respect to $\mathbf{w}$, b for all $i = 1, \cdots, l$. We will prove an associated theorem first, which we will use to prove the convexity of $\mathcal{L}$, $\forall i$.

Theorem 2. *Let $f \colon \mathbb{R}^n \to \mathbb{R}_+$ be a non-negative, real-valued function. Then, $\phi \colon \mathbb{R}^d \to \mathbb{R}$, given by*

$$\phi(\boldsymbol{\theta}) = \int_{\mathbb{R}^n} \max\Big(0, h(\boldsymbol{\theta}, \mathbf{x})\Big) f(\mathbf{x})\, \mathrm{d}\mathbf{x}, \quad (30)$$

is convex with respect to $\boldsymbol{\theta}$ over $\mathbb{R}^d$, if the function h is convex with respect to $\boldsymbol{\theta}$ over $\mathbb{R}^d$.

Proof: Let $\lambda \in [0, 1]$ and $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathbb{R}^d$. Then,

$$\phi(\lambda\boldsymbol{\theta}_1 + (1-\lambda)\boldsymbol{\theta}_2) = \int_{\mathbb{R}^n} \max\Big(0, h(\lambda\boldsymbol{\theta}_1 + (1-\lambda)\boldsymbol{\theta}_2, \mathbf{x}) f(\mathbf{x})\, \mathrm{d}\mathbf{x} \leq \int_{\mathbb{R}^n} \max\Big(0, \lambda h(\boldsymbol{\theta}_1, \mathbf{x})\Big) f(\mathbf{x})\, \mathrm{d}\mathbf{x} + \int_{\mathbb{R}^n} \max\Big(0, (1-\lambda) h(\boldsymbol{\theta}_2, \mathbf{x})\Big) f(\mathbf{x})\, \mathrm{d}\mathbf{x},$$

since h is convex and for $p, q, r \in \mathbb{R}$ it holds that

$$p \leq q + r \Rightarrow \max(0, p) \leq \max(0, q) + \max(0, r).$$

Moreover, $\max(0, \lambda p) = \lambda \max(0, p)$, for $\lambda \geq 0$, $p \in \mathbb{R}$, and thus,

$$\phi(\lambda\boldsymbol{\theta}_1 + (1-\lambda)\boldsymbol{\theta}_2) \leq \lambda \int_{\mathbb{R}^n} \max\Big(0, h(\boldsymbol{\theta}_1, \mathbf{x})\Big) f(\mathbf{x})\, \mathrm{d}\mathbf{x} + (1-\lambda) \int_{\mathbb{R}^n} \max\Big(0, h(\boldsymbol{\theta}_2, \mathbf{x})\Big) f(\mathbf{x})\, \mathrm{d}\mathbf{x} = \lambda\phi(\boldsymbol{\theta}_1) + (1-\lambda)\phi(\boldsymbol{\theta}_2).$$

Consequently, ϕ is convex with respect to $\boldsymbol{\theta}$ over $\mathbb{R}^d$. ■

Using the results of the above theorem, by setting $f(\mathbf{x}) = f_{X_i}(\mathbf{x})$, which is a real-valued, non-negative function (as a probability density function), and $h(\boldsymbol{\theta}, \mathbf{x}) = 1 - y_i(\mathbf{w} \cdot \mathbf{x} + b)$, which is convex with respect to $\boldsymbol{\theta} = (\mathbf{w}^\top, b)^\top$ over $\mathbb{R}^d \equiv \mathbb{R}^n \times \mathbb{R}$, $\mathcal{L}$ is proven to be convex for all i. Consequently, the objective function $\mathcal{J}$ is convex. That means that every local minimum of $\mathcal{J}$ is also a global one.

APPENDIX C
MODELING THE UNCERTAINTY OF AN IMAGE

Let $\mathbf{X} \in \mathbb{R}^n$ be an $r \times r$ image, where $n = r^2$, given in row-wise form as

$$\mathbf{X} = \Big(f_1(\mathbf{t}), \ldots, f_j(\mathbf{t}), \ldots, f_n(\mathbf{t})\Big)^\top \in \mathbb{R}^n, \quad (31)$$

where $f_j\colon \mathbb{R}^2 \to \mathbb{R}$ denotes the intensity function of the j-th pixel, after a translation by $\mathbf{t} = (h, v)^\top$. Fig.5 illustrates this case of study.

We will use Taylor's theorem in order to approximate the intensity function.

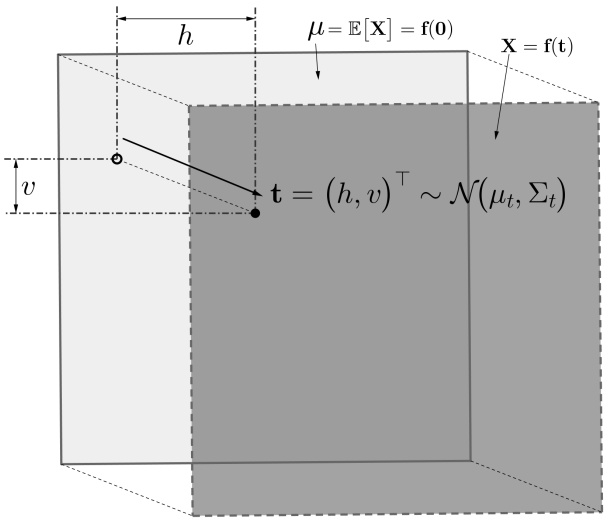


Fig. 5: Image translation by a random vector $\mathbf{t}$.

The multivariate Taylor's theorem [37] is given below without proof.

Theorem 3 (Multivariate Taylor's Theorem)**.** *Let* $\mathbf{t} = (t_1, \ldots, t_n)^\top \in \mathbb{R}^n$ *and consider a function* $f\colon \mathbb{R}^n \to \mathbb{R}$*. Let* $\mathbf{a} = (a_1, \ldots, a_n)^\top \in \mathbb{R}^n$ *and suppose that* f *is differentiable (all first partial derivatives with respect to* $t_1, \ldots, t_n$ *exist) in an open ball* $\mathcal{B}$ *around* $\mathbf{a}$*. Then, the first-order case of Taylor's theorem states that:*

> *If* f *is differentiable on an open ball* $\mathcal{B}$ *around* $\mathbf{a}$ *and* $\mathbf{t} \in \mathcal{B}$*, then*
>
> $$\begin{aligned} f(\mathbf{t}) &= f(\mathbf{a}) + \sum_{k=1}^{n} \frac{\partial f}{\partial t_k}(\mathbf{b})(t_k - a_k) \\ &= f(\mathbf{a}) + \nabla f(\mathbf{b}) \cdot (\mathbf{t} - \mathbf{a}), \end{aligned} \tag{32}$$
>
> *for some* $\mathbf{b}$ *on the line segment joining* $\mathbf{a}$ *and* $\mathbf{b}$*.*

We will use the above theorem in order to approximate the intensity function of the j-th pixel of the given image; i.e., function f_j. That is, around $\mathbf{a}$, the intensity is approximated as follows

$$f_j(\mathbf{t}) \cong f_j(\mathbf{a}) + \nabla f(\mathbf{a}) \cdot (\mathbf{t} - \mathbf{a}),$$

by taking $\mathbf{b}$ to coincide with $\mathbf{a}$. Consequently, by setting $\mathbf{a} = (0, 0)^\top = \mathbf{0}$, the above intensity function is approximated by

$$f_j(\mathbf{t}) = f_j(\mathbf{0}) + \nabla f(\mathbf{0}) \cdot \mathbf{t}.$$

Let us define $\mathbf{f}\colon \mathbb{R}^2 \to \mathbb{R}^n$ given by

$$\mathbf{f}(\mathbf{t}) = \Big(f_1(\mathbf{t}), \ldots, f_j(\mathbf{t}), \ldots, f_n(\mathbf{t})\Big)^\top,$$

then, the image representation can be rewritten as

$$\mathbf{X} = \mathbf{f}(\mathbf{0}) + \begin{pmatrix} \nabla^\top f_1(\mathbf{0}) \\ \vdots \\ \nabla^\top f_j(\mathbf{0}) \\ \vdots \\ \nabla^\top f_n(\mathbf{0}) \end{pmatrix} \mathbf{t}. \tag{33}$$

Let us now assume that $\mathbf{t}$ is a random vector distributed normally with mean $\boldsymbol{\mu}_t$ and covariance matrix Σ_t, i.e. $\mathbf{t} \sim \mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)$. Then, $\mathbf{X}$ is also distributed normally with mean vector and covariance matrix that are given, respectively, by

$$\boldsymbol{\mu} = \mathbb{E}[\mathbf{X}] = \mathbf{f}(\mathbf{0}) + \begin{pmatrix} \nabla^\top f_1(\mathbf{0}) \\ \vdots \\ \nabla^\top f_j(\mathbf{0}) \\ \vdots \\ \nabla^\top f_n(\mathbf{0}) \end{pmatrix} \mathbb{E}[\mathbf{t}], \tag{34}$$

and

$$\begin{aligned} \Sigma &= \mathbb{E}\Big[(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})^\top\Big] \\ &= \begin{pmatrix} \nabla^\top f_1(\mathbf{0}) \\ \vdots \\ \nabla^\top f_j(\mathbf{0}) \\ \vdots \\ \nabla^\top f_n(\mathbf{0}) \end{pmatrix} \mathbb{E}\Big[\mathbf{t}\mathbf{t}^\top\Big] \begin{pmatrix} \nabla^\top f_1(\mathbf{0}) \\ \vdots \\ \nabla^\top f_j(\mathbf{0}) \\ \vdots \\ \nabla^\top f_n(\mathbf{0}) \end{pmatrix}^\top, \end{aligned}$$

or,

$$\Sigma = \begin{pmatrix} \nabla^\top f_1(\mathbf{0}) \\ \vdots \\ \nabla^\top f_j(\mathbf{0}) \\ \vdots \\ \nabla^\top f_n(\mathbf{0}) \end{pmatrix} \Sigma_t \begin{pmatrix} \nabla^\top f_1(\mathbf{0}) \\ \vdots \\ \nabla^\top f_j(\mathbf{0}) \\ \vdots \\ \nabla^\top f_n(\mathbf{0}) \end{pmatrix}^\top. \tag{35}$$

Thus, by setting $\mathbf{t} \sim \mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)$, it holds that $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, where the mean vector $\boldsymbol{\mu}$ and the covariance matrix Σ are given by (34) and (35), respectively.

ACKNOWLEDGMENT

This work was supported by the European Commission under contracts FP7-287911 LinkedTV and FP7-600826 ForgetIT.

REFERENCES

[1] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*. ACM, 1992, pp. 144–152.

[2] V. N. Vapnik and A. J. Chervonenkis, "Theory of pattern recognition," 1974.

[3] V. N. Vapnik, "Statistical learning theory (adaptive and learning systems for signal processing, communications and control series)," 1998.

[4] A. J. Smola, B. Schölkopf, and K.-R. Müller, "The connection between regularization operators and support vector kernels," *Neural networks*, vol. 11, no. 4, pp. 637–649, 1998.

[5] T. Evgeniou, M. Pontil, and T. Poggio, "Regularization networks and support vector machines," *Advances in Computational Mathematics*, vol. 13, no. 1, pp. 1–50, 2000.

[6] P. L. Bartlett and S. Mendelson, "Rademacher and gaussian complexities: Risk bounds and structural results," *The Journal of Machine Learning Research*, vol. 3, pp. 463–482, 2003.

[7] J. Bi and T. Zhang, "Support vector classification with input data uncertainty." in *NIPS*, 2004.

[8] F. Alizadeh and D. Goldfarb, "Second-order cone programming," *Mathematical programming*, vol. 95, no. 1, pp. 3–51, 2003.

[9] A. Ben-Tal and A. Nemirovski, "Robust convex optimization," *Mathematics of Operations Research*, vol. 23, no. 4, pp. 769–805, 1998.

[10] D. Bertsimas, D. B. Brown, and C. Caramanis, "Theory and applications of robust optimization," *SIAM review*, vol. 53, no. 3, pp. 464–501, 2011.

[11] G. R. Lanckriet, L. E. Ghaoui, C. Bhattacharyya, and M. I. Jordan, "A robust minimax approach to classification," *The Journal of Machine Learning Research*, vol. 3, pp. 555–582, 2003.

[12] A. W. Marshall, I. Olkin *et al.*, "Multivariate chebyshev inequalities," *The Annals of Mathematical Statistics*, vol. 31, no. 4, pp. 1001–1014, 1960.

[13] P. K. Shivaswamy, C. Bhattacharyya, and A. J. Smola, "Second order cone programming approaches for handling missing and uncertain data," *The Journal of Machine Learning Research*, vol. 7, pp. 1283–1314, 2006.

[14] C. Bhattacharyya, P. K. Shivaswamy, and A. J. Smola, "A second order cone programming formulation for classifying missing data." in *NIPS*, 2004.

[15] H. Xu, C. Caramanis, and S. Mannor, "Robustness and regularization of support vector machines," *The Journal of Machine Learning Research*, vol. 10, pp. 1485–1510, 2009.

[16] H. Xu and S. Mannor, "Robustness and generalization," *Machine learning*, vol. 86, no. 3, pp. 391–423, 2012.

[17] Z. Qi, Y. Tian, and Y. Shi, "Robust twin support vector machine for pattern classification," *Pattern Recognition*, vol. 46, no. 1, pp. 305–316, 2013.

[18] O. L. Mangasarian and E. W. Wild, "Multisurface proximal support vector machine classification via generalized eigenvalues," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 1, pp. 69–74, 2006.

[19] R. Khemchandani, S. Chandra *et al.*, "Twin support vector machines for pattern classification," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 5, pp. 905–910, 2007.

[20] F. De La Torre and M. J. Black, "A framework for robust subspace learning," *International Journal of Computer Vision*, vol. 54, no. 1-3, pp. 117–142, 2003.

[21] S. Liwicki, S. Zafeiriou, G. Tzimiropoulos, and M. Pantic, "Efficient online subspace learning with an indefinite kernel for visual tracking and recognition," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 23, no. 10, pp. 1624–1636, 2012.

[22] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Uncorrelated multilinear principal component analysis for unsupervised multilinear subspace learning," *Neural Networks, IEEE Transactions on*, vol. 20, no. 11, pp. 1820–1836, 2009.

[23] M. V. Jankovic and H. Ogawa, "Modulated hebb-oja learning rule-a method for principal subspace analysis," *Neural Networks, IEEE Transactions on*, vol. 17, no. 2, pp. 345–356, 2006.

[24] T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu, "The entire regularization path for the support vector machine," *Ann Arbor*, vol. 1001, pp. 48 109–1092, 2004.

[25] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[26] C. G. Broyden, "The convergence of a class of double-rank minimization algorithms 1. general considerations," *IMA Journal of Applied Mathematics*, vol. 6, no. 1, pp. 76–90, 1970.

[27] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[28] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, W. Kraaij, A. F. Smeaton, and G. Quenot, "Trecvid 2014 – an overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proceedings of TRECVID 2014*. NIST, USA, 2014.

[29] F. Gkalelis, Nikolaos andMarkatopoulou, A. Moumtzidou, D. Galanopoulos, K. Avgerinakis, N. Pittaras, S. Vrochidis, V. Mezaris, I. Kompatsiaris, and I. Patras, "Iti-certh participation to trecvid 2014," in *Proceedings TRECVID Workshop*, 2014.

[30] S. Robertson, "A new interpretation of average precision," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2008, pp. 689–690.

[31] N. Gkalelis, V. Mezaris, I. Kompatsiaris, and T. Stathaki, "Mixture subclass discriminant analysis link to restricted gaussian model and other generalizations," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 24, no. 1, pp. 8–21, 2013.

[32] S. Zafeiriou, G. Tzimiropoulos, M. Petrou, and T. Stathaki, "Regularized kernel discriminant analysis with a robust kernel for face recognition and verification," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 23, no. 3, pp. 526–534, 2012.

[33] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947.

[34] Y. LeCun and C. Cortes, "The mnist database of handwritten digits," 1998.

[35] A. Ghio, D. Anguita, L. Oneto, S. Ridella, and C. Schatten, "Nested sequential minimal optimization for support vector machines," in *Artificial Neural Networks and Machine Learning–ICANN 2012*. Springer, 2012, pp. 156–163.

[36] W. W. Hines, D. C. Montgomery, and D. M. G. C. M. Borror, *Probability and statistics in engineering*. John Wiley & Sons, 2008.

[37] T. M. Apostol, "Calculus. 1967," *Jon Wiley & Sons*, 1967.

Christos Tzelepis Christos Tzelepis received the Diploma degree in electrical engineering from Aristotle University of Thessaloniki, Greece, in 2011. During his diploma thesis, he focused on machine learning techniques with training data of variable reliability. Currently, he is a PhD student in Electronic Engineering and Computer Science at Queen Mary, University of London, within the field of discriminative machine learning, and works as a research assistant at the Information Technologies Institute (ITI) of the Centre of Research & Technology Hellas (CERTH).

Vasileios Mezaris Vasileios Mezaris received the BSc and PhD in Electrical and Computer Engineering from the Aristotle University of Thessaloniki in 2001 and 2005, respectively. He is a Senior Researcher (Researcher B) at the Information Technologies Institute (ITI) of the Centre for Research of Technology Hellas (CERTH). His research interests include image and video analysis, retrieval, event detection, machine learning for multimedia analysis. He is an Associate Editor for the IEEE Trans. on Multimedia and a Senior Member of the IEEE.

Ioannis Patras Ioannis Patras received the B.Sc. and M.Sc. degrees in computer science from the Computer Science Department, University of Crete, Heraklion, Greece, in 1994 and 1997, respectively, and the Ph.D. degree from the Department of Electrical Engineering, Delft University of Technology, Delft (TU Delft), The Netherlands, in 2001. He is a Senior Lecturer in computer vision with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London, U.K. His current research interests are in the area of computer vision and pattern recognition, with emphasis on the analysis of human motion, including the detection, tracking, and understanding of facial and body gestures motion analysis and in applications in multimedia data management, multimodal human computer interaction, and visual communication. He is an Associate Editor of the Image and Vision Computing Journal.